IEEE WCCI 2020
IEEE World Congress on Computational Intelligence
Virtual Conference – July 19-24, 2020

Evolutionary Fuzzy Markup Language-based Fuzzy Inference Systems with Feature Selection for Winning Rate Prediction in Game of Go

Yuichi Omozaki[1], Naoki Masuyama[1], Yusuke Nojima[1], and Hisao Ishibuchi[2]
[1] Osaka Prefecture University, [2] Southern University of Science and Technology

Overview

1. Introduction

2. Our Proposed Method
    - General Framework of Evolutionary FML-based Fuzzy Inference System
    - Rule Base Optimization
    - Knowledge Base Optimization
    - Feature Selection

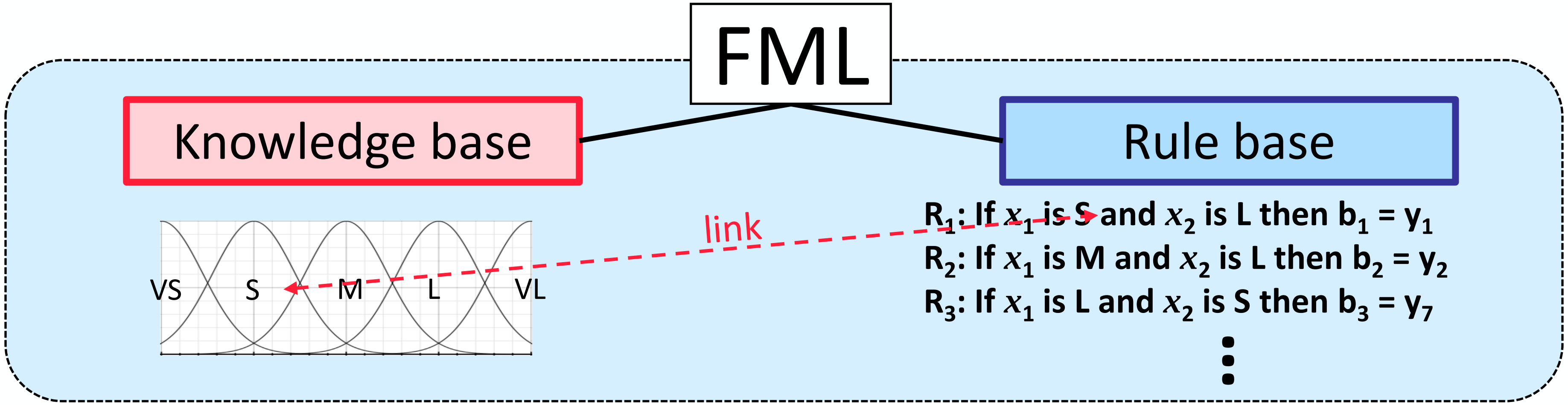
3. Computational Experiments

4. Conclusion

Introduction

Fuzzy Markup Language (FML)

- Fuzzy Markup Language (FML) is composed of the knowledge base (KB) and the rule base (RB).
- KB defines the shapes of the membership functions which represent fuzzy sets.
- RB is composed of a set of fuzzy if-then rules.
- Researchers can describe various fuzzy systems independent of the software.



Introduction

FML-based Machine Learning Competition

- 60 game results of AlphaGo Master series are provided.
- The goal of this competition is to design accurate and interpretable fuzzy rule-based inference system using FML.
- The datasets from Master Game 1 to Game 45 are used as the training data, while the remaining datasets from Game 46 to Game 60 are set to be the test data.

Each pattern has six attributes and one output.

DBSN(t - 1)	DBSN(t)	The number of simulations for Black predicted by Darkforest AI.
DWSN(t - 1)	DWSN(t)	The number of simulations for White predicted by Darkforest AI.
DBWR(t - 1)	DBWR(t)	The win rate of Black predicted by Darkforest AI.
DWWR(t - 1)	DWWR(t)	The win rate of White predicted by Darkforest AI.
DBTMR(t - 1)	DBTMR(t)	The top-move-rate of Black predicted by Darkforest AI.
DWTMR(t - 1)	DWTMR(t)	The top-move-rate of White predicted by Darkforest AI.
EBWR(t)		The present win rate of Black predicted by ELF OpenGo AI.
DBWR(t + 1)		The future win rate of Black predicted by Darkforest AI

Available:
https://github.com/CI-labo-OPU/FML_Competition2020/Comp2020Data

Our proposed Method

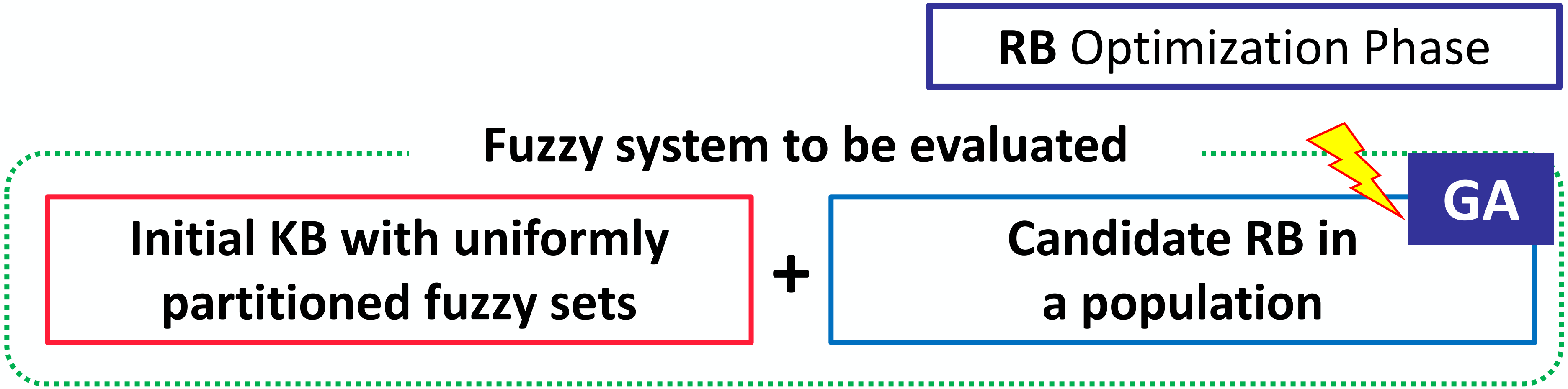
Evolutionary FML-based Fuzzy Inference System

- We design a fuzzy inference system based on FML using a Genetic Algorithm (GA) for two optimization phases.

➢ First, we optimize the RB by GA with the initial KB.

➢ Then, we optimize the KB by GA with the best RB in the previous phase.

➢ After that, we optimize the RB again with the best KB found so far.



Our proposed Method

Evolutionary FML-based Fuzzy Inference System

- We design a fuzzy inference system based on FML using Genetic Algorithm (GA) composed of two optimization phases.

➢First, we optimize the RB by GA with initial KB.

➢Then, we optimize the KB by GA with the best RB in the previous phase.

➢After that, we optimize the RB again with the best KB found so far.

KB Optimization Phase

GA

Fuzzy system to be evaluated

Candidate KB in a population **+** **The best RB found in the previous phase**

Our proposed Method

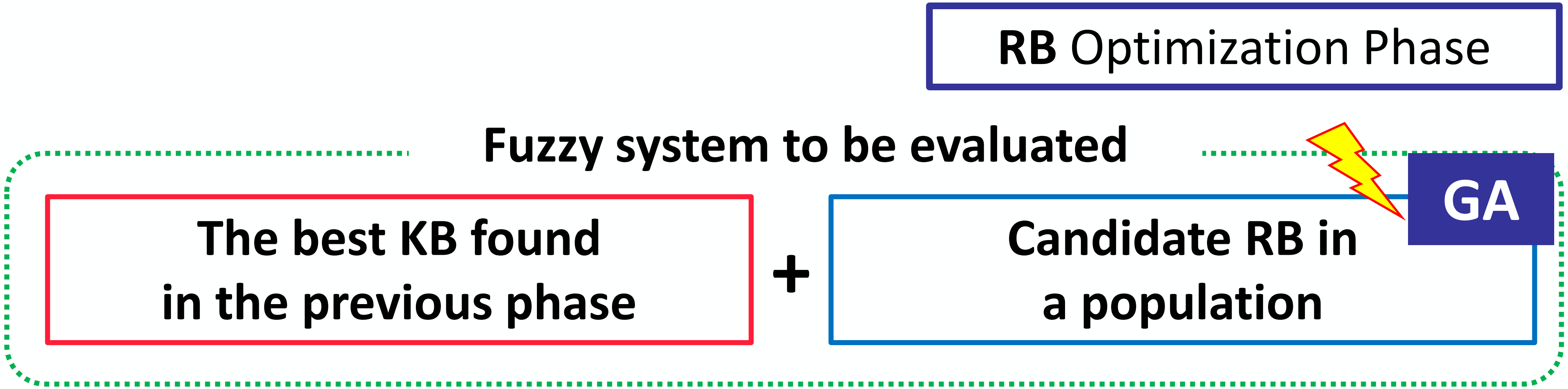
Evolutionary FML-based Fuzzy Inference System

- We design a fuzzy inference system based on FML using Genetic Algorithm (GA) composed of two optimization phases.

➢First, we optimize the RB by GA with initial KB.

➢Then, we optimize the KB by GA with the best RB in the previous phase.

➢After that, we optimize the RB again with the best KB found so far.



Our proposed Method

Evolutionary FML-based Fuzzy Inference System

- We design a fuzzy inference system based on FML using Genetic Algorithm (GA) composed of two optimization phases.

Two optimization phases are alternately performed with the best KB/RB optimized in the previous phase.

KB Optimization Phase ⇄ **RB** Optimization Phase

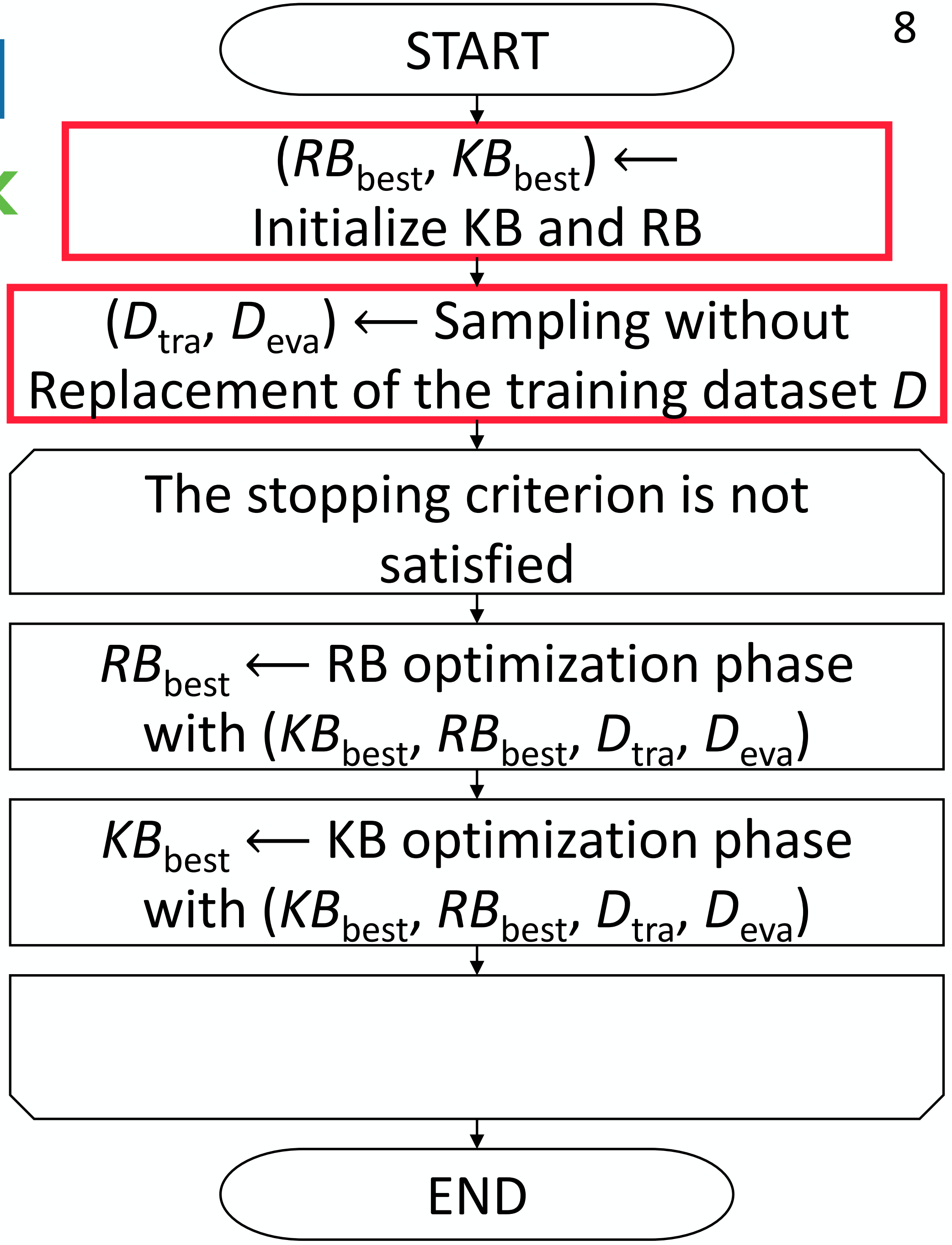
Fuzzy system to be evaluated

The optimized KB using the best RB found so far + **The optimized RB using the best KB found so far**

Our proposed Method

General Algorithm Framework

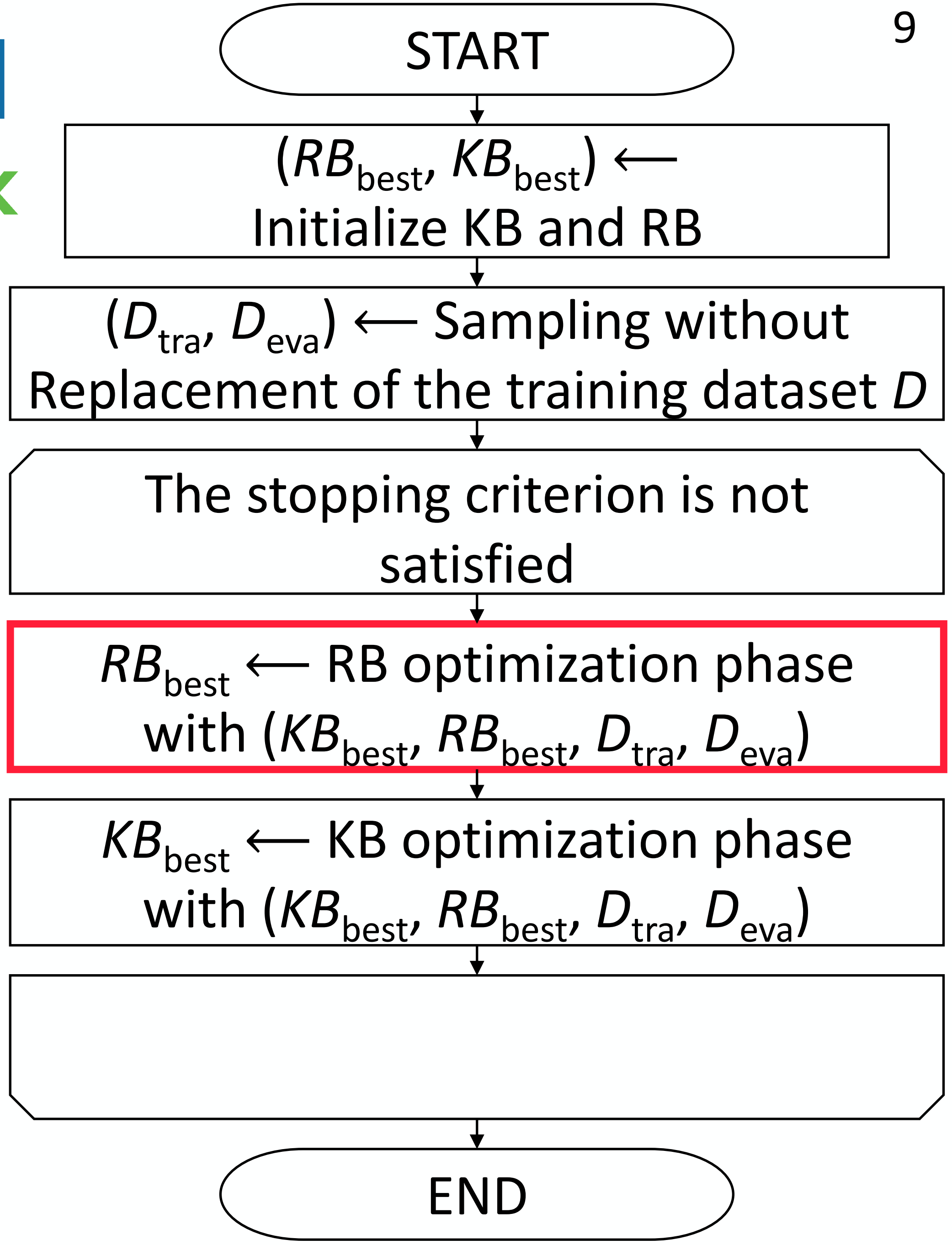
1. Initialize KB and RB
   - Initial KB is defined with the initial fuzzy sets
   - Initial RB is defined with empty set
2. Sampling Evaluation Dataset D_{eva}
3. RB Optimization Phase
4. KB Optimization Phase
5. Repeat Step 3 and Step 4.



Our proposed Method

General Algorithm Framework

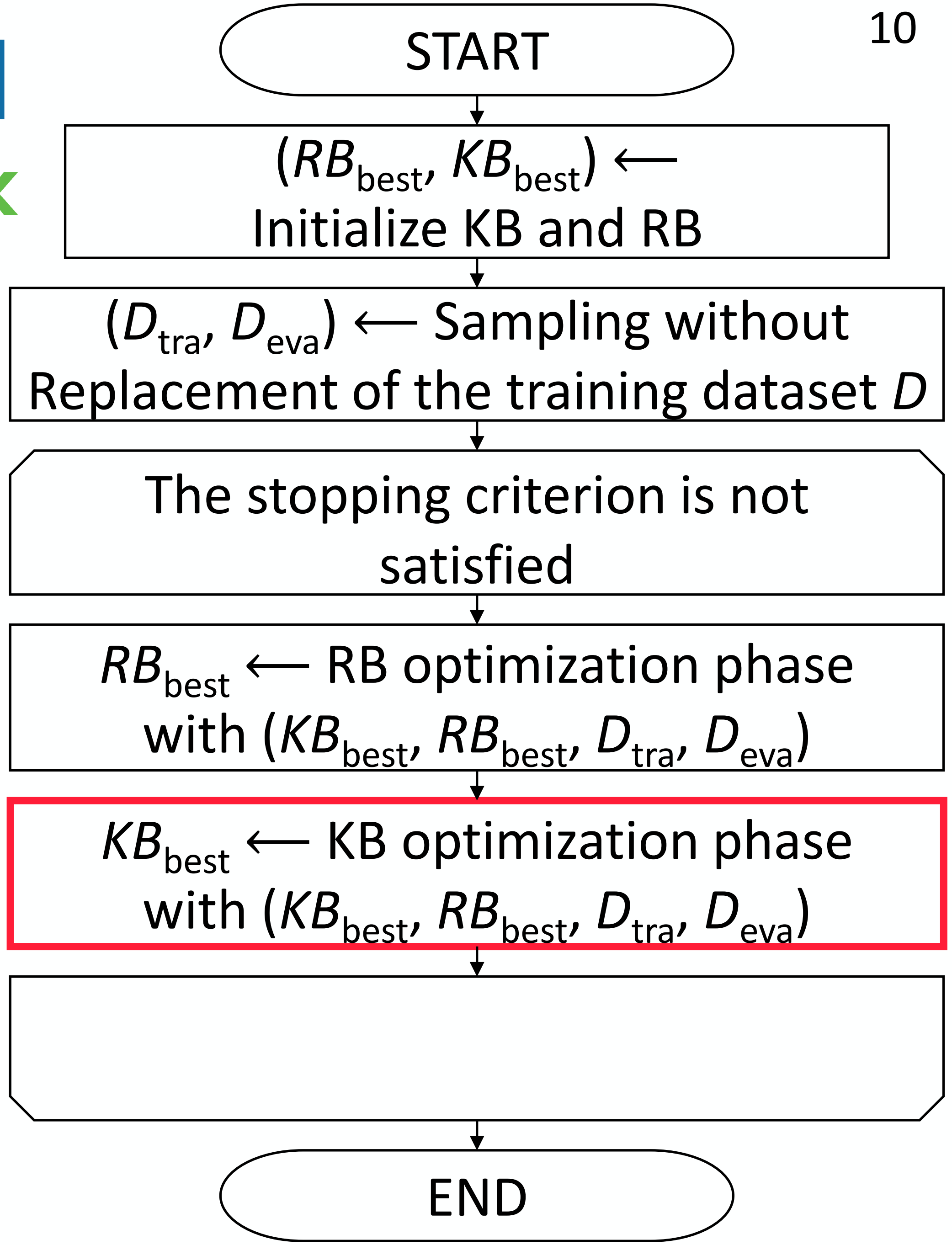
1. Initialize KB and RB
2. Sampling Evaluation Dataset D_{eva}
3. RB Optimization Phase
   - I. Initialize a population by heuristic rule generation method
   - II. Evaluate the initial population
   - III. Evolve the population
     - i. Offspring generalization and mutation
     - ii. Rule Pruning
     - iii. Evaluation of the offspring population
     - iv. Population update
4. KB Optimization Phase
5. Repeat Step 3 and Step 4.



Our proposed Method

General Algorithm Framework

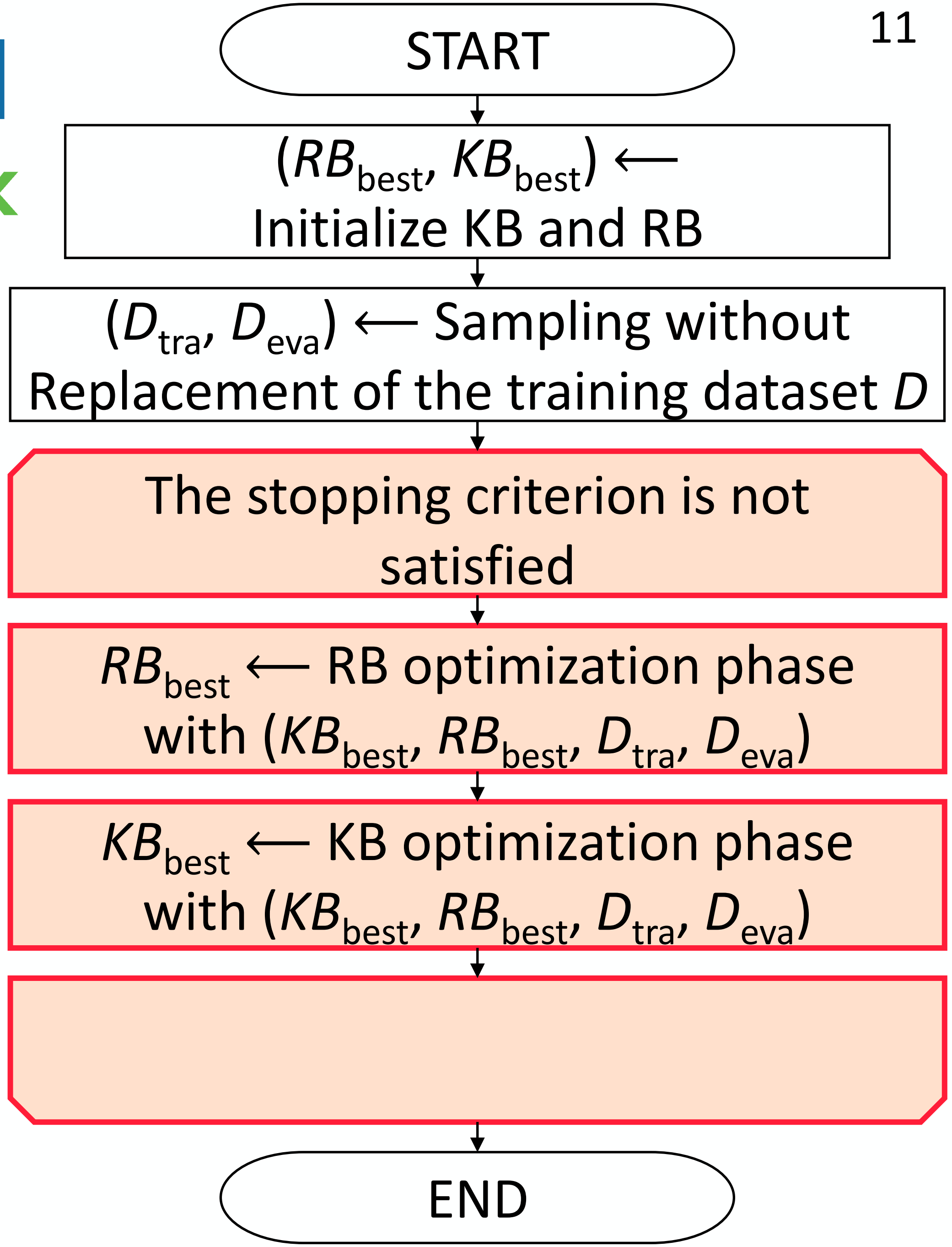
1. Initialize KB and RB
2. Sampling Evaluation Dataset D_{eva}
3. RB Optimization Phase
4. KB Optimization Phase
   I. Initialize a population by perturbation to the current best KB
   II. Evaluate the initial population
   III. Evolve the population
      i. Offspring generalization and mutation
      ii. Rule Pruning.
      iii. Evaluation of the offspring population.
      iv. Population update.
5. Repeat Step 3 and Step 4.



Our proposed Method

General Algorithm Framework

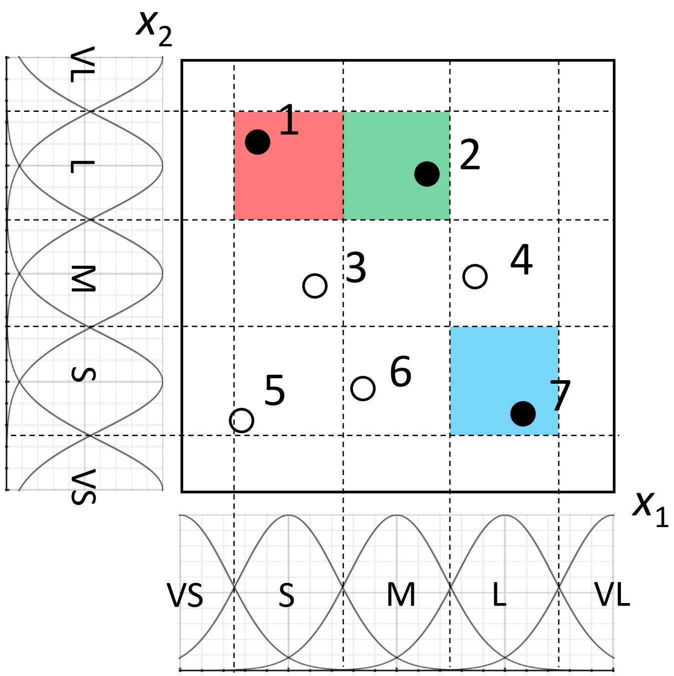
1. Initialize KB and RB
2. Sampling Evaluation Dataset D_{eva}
3. RB Optimization Phase
4. KB Optimization Phase
5. Repeat Step 3 and Step 4.



Rule Base Optimization Phase

Population Initialization

- Initial rule sets are generated by a heuristic rule generation method.
- In the heuristic rule generation method, each fuzzy if-then rule is generated for covering a randomly selected pattern.



If patterns 1, 2, and 7 are randomly selected, the following three rules are generated.

R$_1$: If x_1 is S and x_2 is L then $b_1 = y_1$

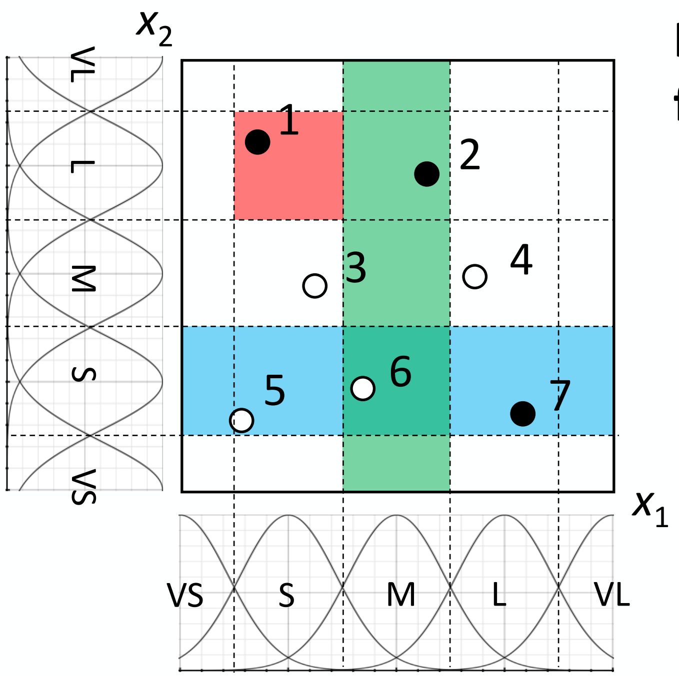
R$_2$: If x_1 is M and x_2 is L then $b_2 = y_2$

R$_3$: If x_1 is L and x_2 is S then $b_3 = y_7$

Rule Base Optimization Phase

Heuristic Rule Generation Method

- A fuzzy if-then rule is generated for covering a randomly selected pattern.
- The most compatible fuzzy set is used for each attribute. The consequent output is specified by that of the selected pattern.



If patterns 1, 2, and 7 are randomly selected, the following three rules are generated.

R_1: If x_1 is S and x_2 is L then $b_1 = y_1$

R_2: If x_1 is M and x_2 is DC then $b_2 = y_2$

R_3: If x_1 is DC and x_2 is S then $b_3 = y_7$

Some attribute conditions are replaced with *don't care* to make the rule generalized.

Rule Base Optimization Phase

Evaluation of Population

1. The nth individual calculates the consequent part C_n with the learning set:

$$C_n \leftarrow C_n + \eta\,(T_k - O_k)\frac{\mu_n(\boldsymbol{x}_k)}{\sum_{i=1}^{N_R}\mu_i(\boldsymbol{x}_k)}$$

Here, η is the learning rate, T_k is the desired output of kth learning pattern x_k, O_k is the output predicted by fuzzy inference system with kth learning pattern, and $\mu_n(\cdot)$ is membership value of the nth rule for a learning pattern.

2. An Mean Squared Error (MSE) is calculated as a fitness for the evaluation set:

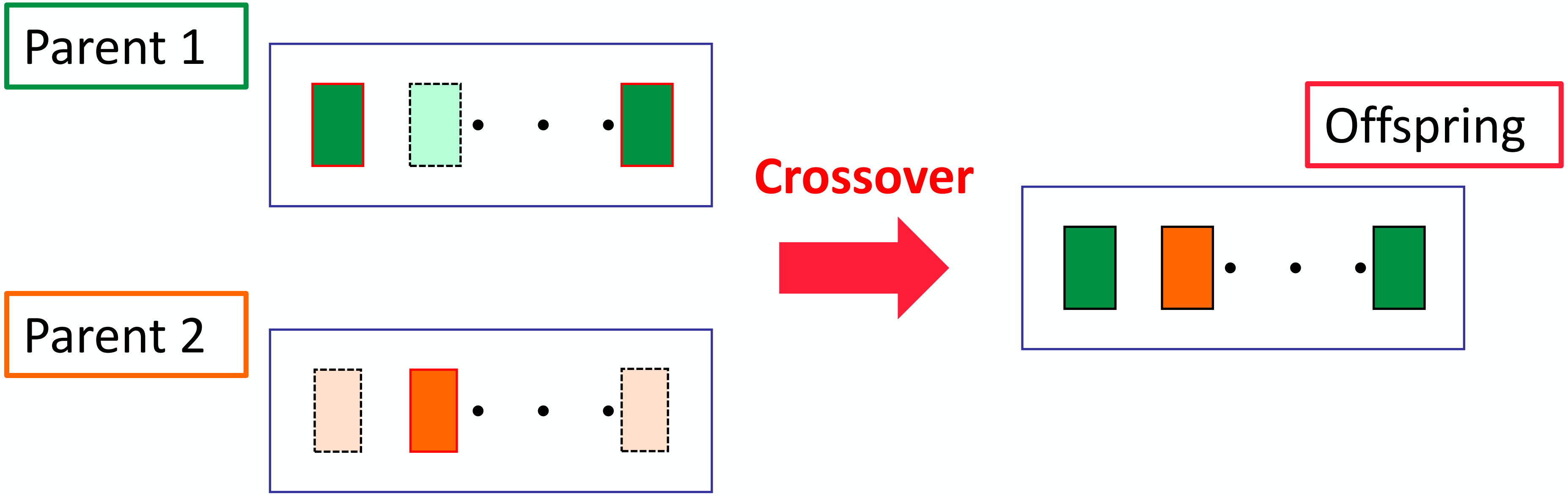
$$MSE = \frac{1}{M}\sum_{i=1}^{M}(O_i - T_i)^2$$

Here, M is the size of the dataset, i is the index of the pattern.

Rule Base Optimization Phase

Crossover and Mutation

- A new individual inherits some rules at random from two parents selected by Binary Tournament Selection.
- The mutation is applied according to the mutation rate.
- One fuzzy set of the antecedent part is randomly replaced with another fuzzy set or “don’t care”.



Rule Base Optimization Phase

Crossover and Mutation

- A new individual inherits some rules at random from two parents selected by Binary Tournament Selection.
- The mutation is applied according to the mutation rate.
- One fuzzy set of the antecedent part is randomly replaced with another fuzzy set or “don’t care”.

R_1: If x_1 is S and x_2 is L then …
R_2: If x_1 is M and x_2 is L then …

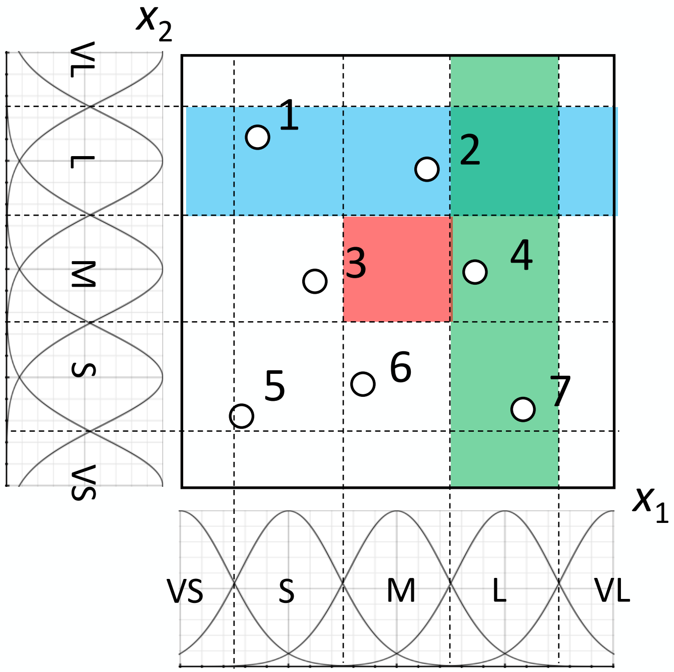
⬇ **Mutation**

R_1: If x_1 is M and x_2 is L then …
R_2: If x_1 is M and x_2 is DC then …

Rule Base Optimization Phase

Rule Pruning

- After genetic operations, the compatibility of each rule in a new offspring model is examined with the training data.
- Rules which do not cover any patterns having more than the compatibility grade of 0.5 are removed from the offspring model.



Assume that a new model with the following rules is generated by genetic operations.

R_1: If x_1 is M and x_2 is M then b_1

R_2: If x_1 is L and x_2 is DC then b_2

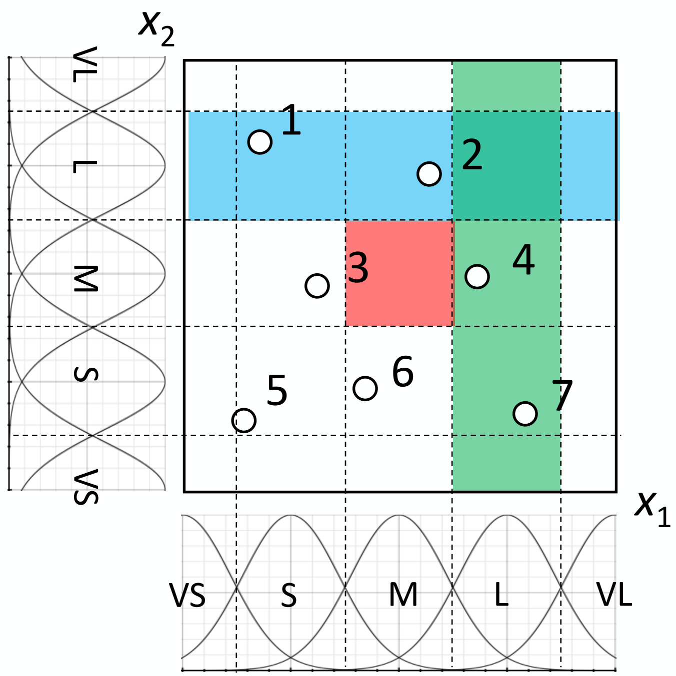
R_3: If x_1 is DC and x_2 is L then b_3

R_1 does not cover any patterns having more than the compatibility grade of 0.5.

Rule Base Optimization Phase

Rule Pruning

- After genetic operations, the compatibility of each rule in a new offspring model is examined for training data.
- Rules which do not cover any patterns with more than 0.5 degree are removed from the offspring model.



Assume that a new model with the following rules is generated by genetic operations.

~~**R_1: If x_1 is M and x_2 is M then b_1**~~

R_2: If x_1 is L and x_2 is DC then b_2

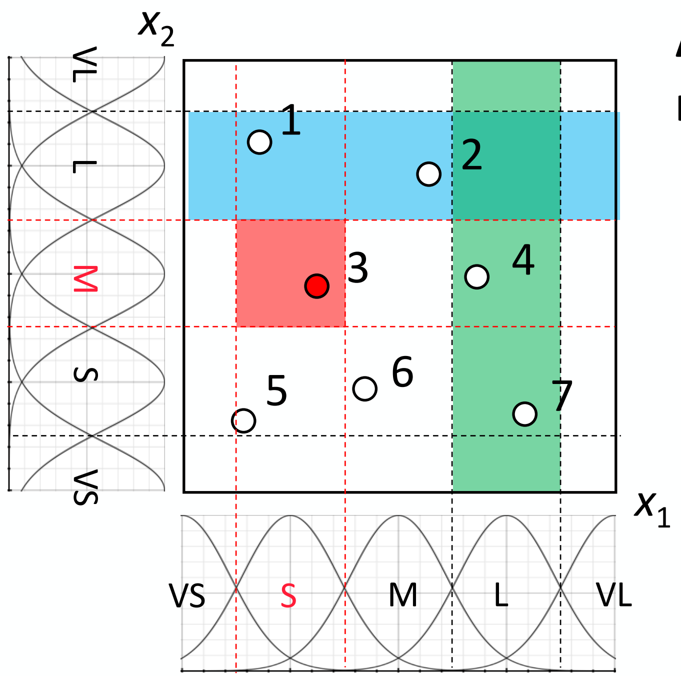
R_3: If x_1 is DC and x_2 is L then b_3

R_1 does not cover any patterns with more than 0.5 degree. => R_1 is removed.

Rule Base Optimization Phase

Rule Addition by Heuristic Rule Generation

- After genetic operations, the compatibility of each rule in a new offspring model is examined for training data.
- Rules which do not cover any patterns with more than 0.5 degree are removed from the offspring model.



A pattern which is not well covered by any rule is randomly selected for each new rule.

R_1: If x_1 is S and x_2 is M then b_1

R_2: If x_1 is L and x_2 is DC then b_2

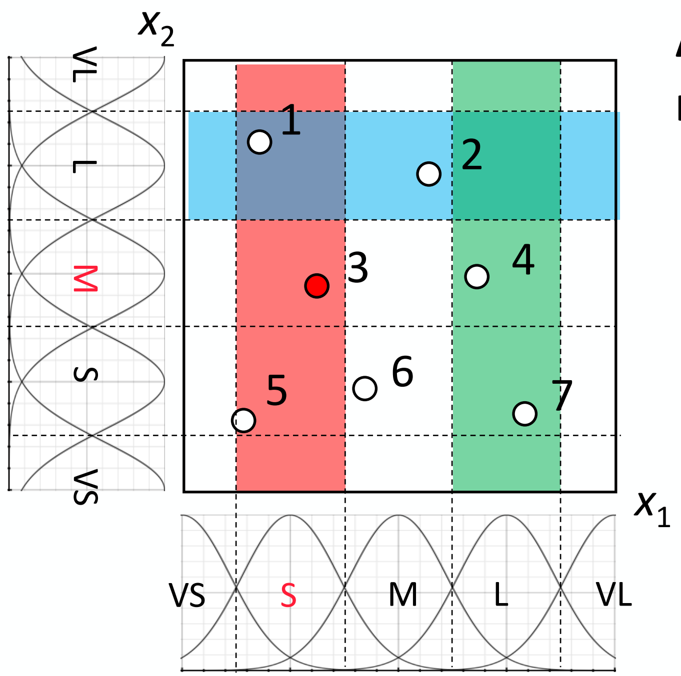
R_3: If x_1 is DC and x_2 is L then b_3

New rules are generated in a heuristic manner.

Rule Base Optimization Phase

Rule Addition by Heuristic Rule Generation

- After genetic operations, the compatibility of each rule in a new offspring model is examined for training data.
- Rules which do not cover any patterns with more than 0.5 degree are removed from the offspring model.



A pattern which is not well covered by any rule is randomly selected for each new rule.

R_1: If x_1 is S and x_2 is DC then b_1

R_2: If x_1 is L and x_2 is DC then b_2

R_3: If x_1 is DC and x_2 is L then b_3

Some attribute conditions are replaced with *don't care* to make the rule generalized.

Rule Base Optimization Phase

Population Update in RB Phase

- The individuals with better fitness values (i.e., lower error rates) are selected from the current and offspring populations in order to make the new population for the next generation.
- If the termination condition is not satisfied, then continue to generate offspring models.
- Otherwise, the best RB in the final generation is selected as an output of the RB optimization phase.

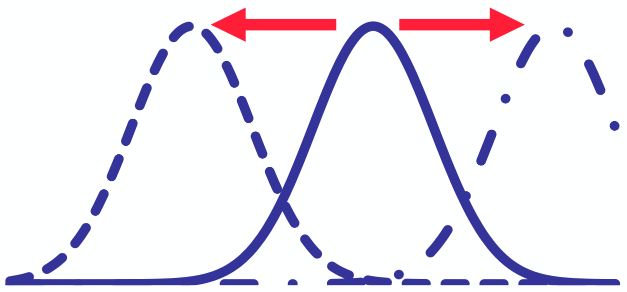
Knowledge Base Optimization Phase

Knowledge Base Population Initialization

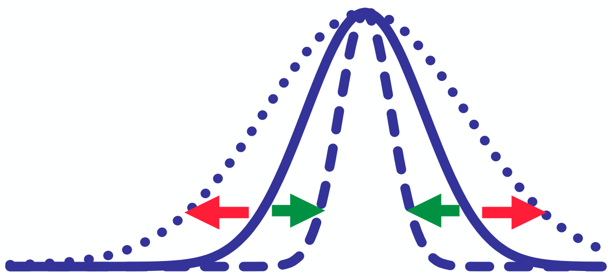
- A candidate KB is generated by perturbing to the shapes of membership functions belong to the best RB obtained in the previous phase.
- A small bias is added to each parameter of membership functions (i.e., the mean and the variance)

The perturbing method for Gaussian shape

Perturbing to the mean



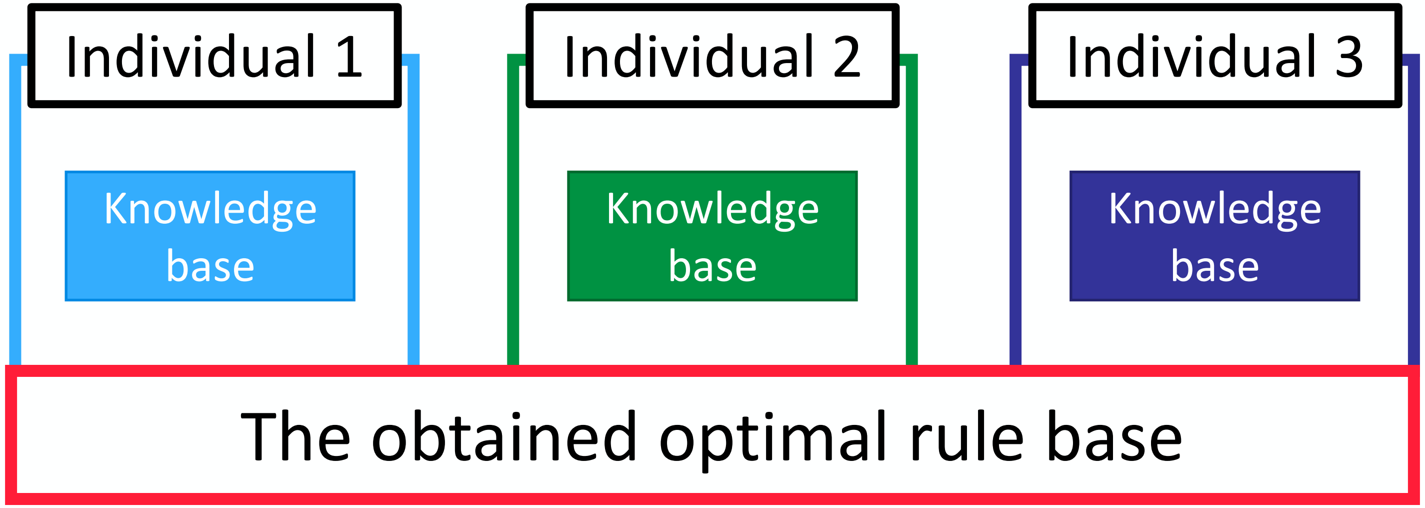
Perturbing to the variance



Knowledge Base Optimization Phase

Evaluation of the Knowledge Base

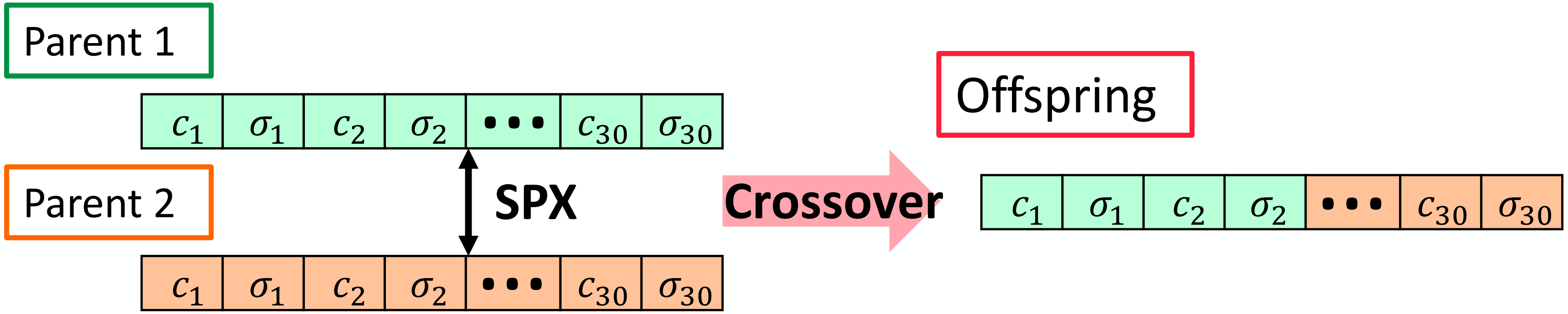
- Each individual consists of a candidate KB and the best RB obtained in the previous phase.
- An MSE is calculated as a fitness value for evaluation dataset using the candidate KB and the best RB.



Knowledge Base Optimization Phase

Crossover and Perturbation

- Each fuzzy set is represented by two parameters (i.e., the mean and the variance).
- The number of decision variables in the individual is 2 times that of a fuzzy sets in KB.
- A new individual is generated by Single Point Crossover from two parents selected by Binary Tournament Selection.
- However, the crossover point is always set to be a number multiple of two.



Knowledge Base Optimization Phase

Crossover and Mutation

- Each fuzzy set is represented by two parameters (i.e., the mean and the variance).
- The number of decision variables in the individual is 2 times that of a fuzzy sets in KB.
- A new individual is generated by Single Point Crossover from two parents selected by Binary Tournament Selection.
- However, the crossover point is always set to be a number multiple of two.
- The mutation is performed by the perturbation method.

Offspring

c_1	σ_1	c_2	σ_2	•••	c_{30}	σ_{30}

Mutation

Offspring

c_1	σ_1	$\boxed{c_2{}'}$	$\boxed{\sigma_2{}'}$	•••	c_{30}	σ_{30}

Knowledge Base Optimization Phase

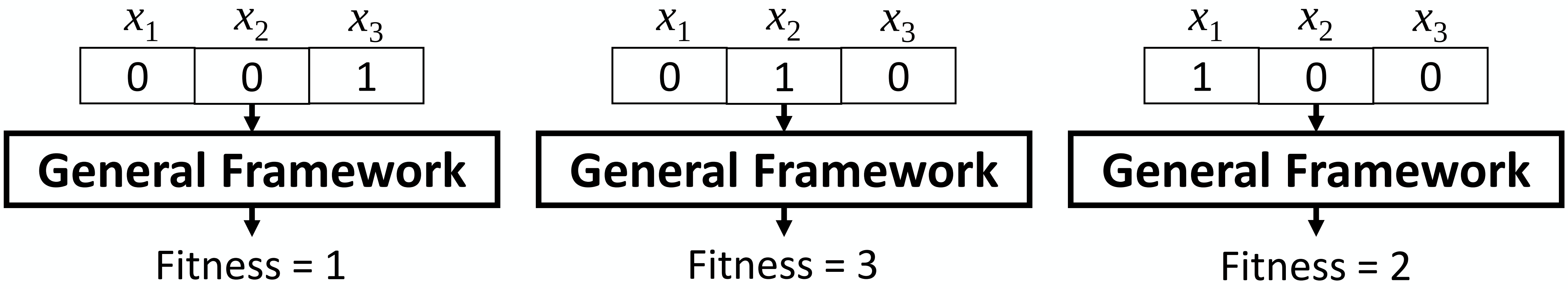
Population Update

- The individuals with better fitness values (i.e., lower error rates) are selected from the current and offspring populations in order to make the new population in the next generation.
- If the termination condition is not satisfied, then continue to generate offspring.
- Otherwise, the best KB in the final generation is selected as an output in the KB optimization phase.

Feature Selection

Sequential Forward Selection

- We apply a sequential forward selection method[1] to our proposed method.
- The selected features are represented by a mask of feature array.
- First, using one feature, we execute our proposed general framework and evaluate a fuzzy inference system using the features.

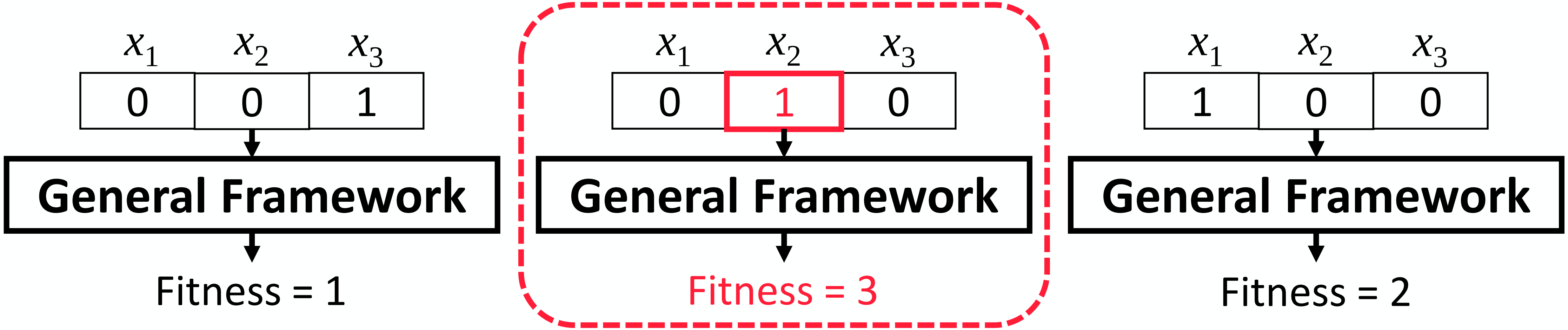


[1] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119-1125, 1994.

Feature Selection

Sequential Forward Selection

- We apply a sequential forward selection method[1] to our proposed method.
- The selected features are represented by a mask of feature array.
- Then, a combination of features used in the best fuzzy inference system is selected by the fitness values.

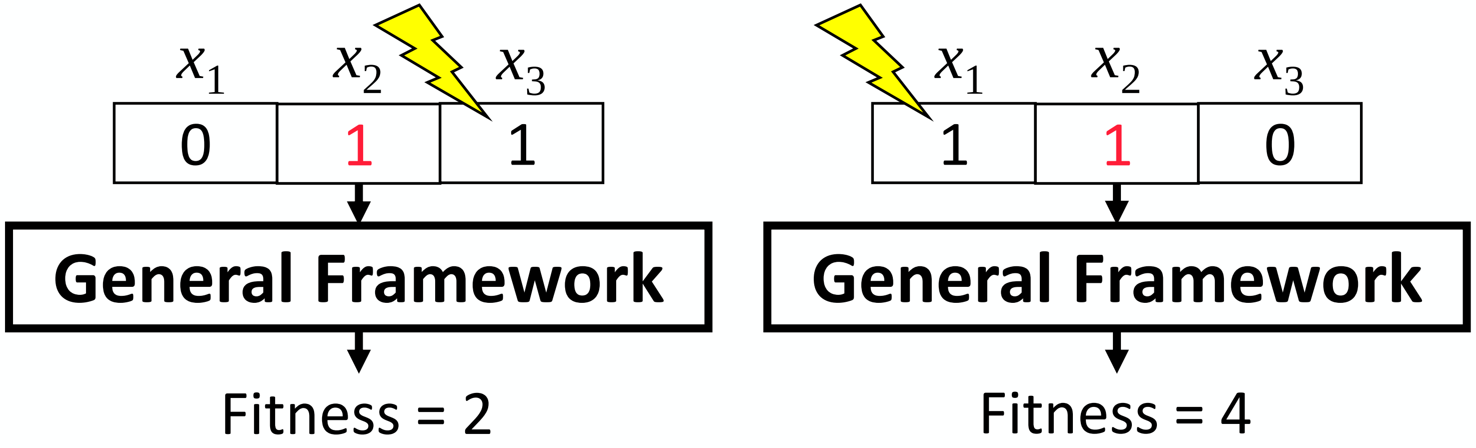


[1] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119-1125, 1994.

Feature Selection

Sequential Forward Selection

- We apply a sequential forward selection method[1] to our proposed method.
- The selected features are represented by a mask of feature array.
- Next, the other feature is added to the mask and the masks of selected features are evaluated in the same manner.

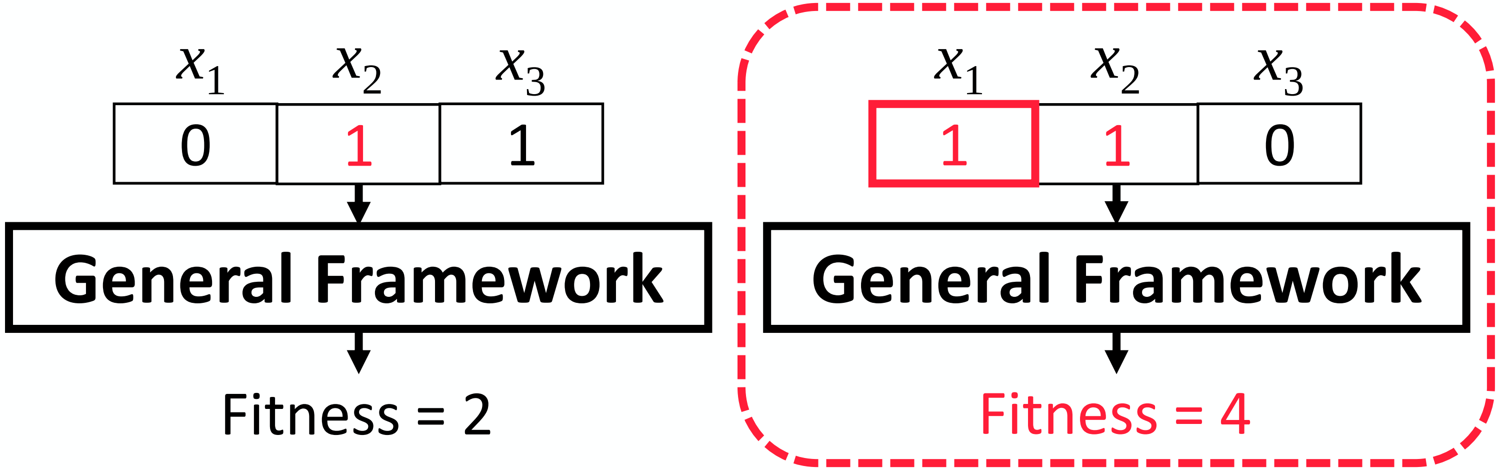


[1] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119-1125, 1994.

Feature Selection

Sequential Forward Selection

- We apply a sequential forward selection method[1] to our proposed method.
- The selected features are represented by a mask of feature array.
- Repeat that the other feature is added to the mask and the masks of selected features are evaluated in the same manner.



[1] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119-1125, 1994.

Computational Experiments

Experimental Settings

For General Framework

- Learning rate (η): **0.5**
- Epoch to calculate consequent part: **100**
- Crossover rate for both phases: **0.9**
- Size of the evaluation dataset: **500**
- Range of the number of rules per rule set: **[500, 5,000]**
- The number of generation: **20**
- Population size: **20**

For Rule Base Optimization Phase

- Mutation rate: **0.1**

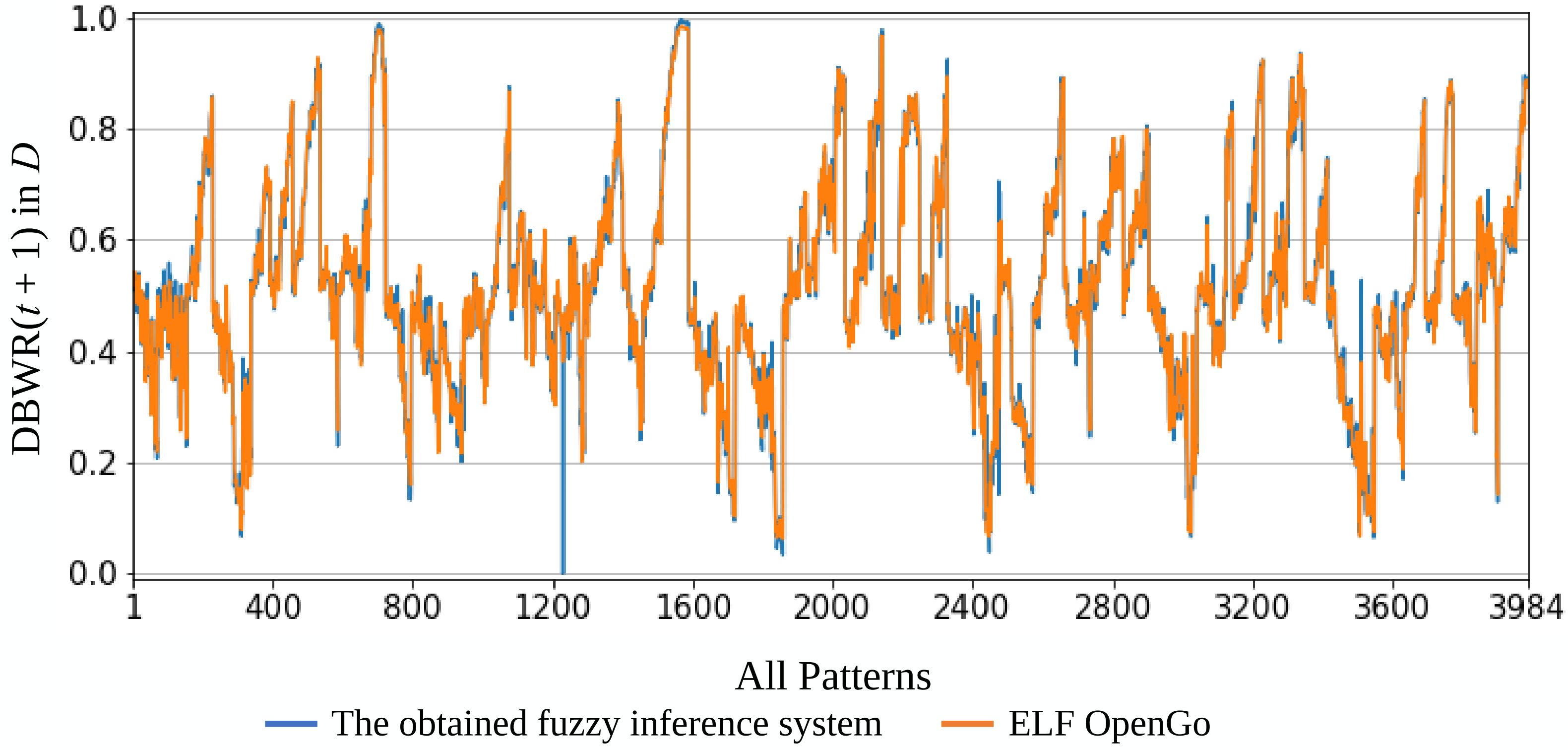
For Knowledge Base Optimization Phase

- Perturbation rate: **0.8**

Computational Experiments

Inferred DBWR(t+1) Curves for the Training Dataset

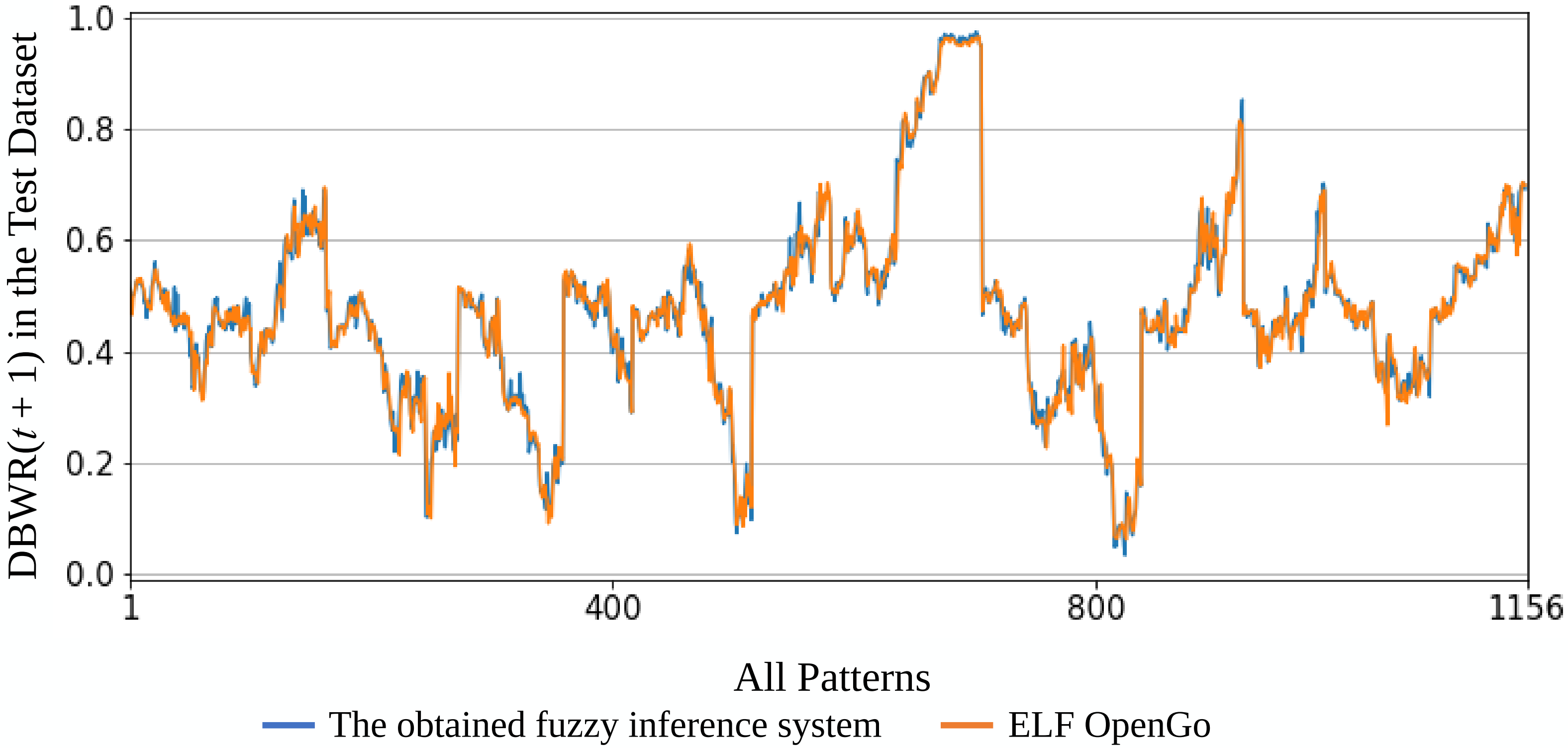
- This figure shows that inferred DBWR(t+1) curves for the training dataset by our proposed method and ELF OpenGo AI.
- Note that the training dataset includes the learning set and the evaluation set.



Computational Experiments

Inferred DBWR(t+1) Curves for the Test Dataset

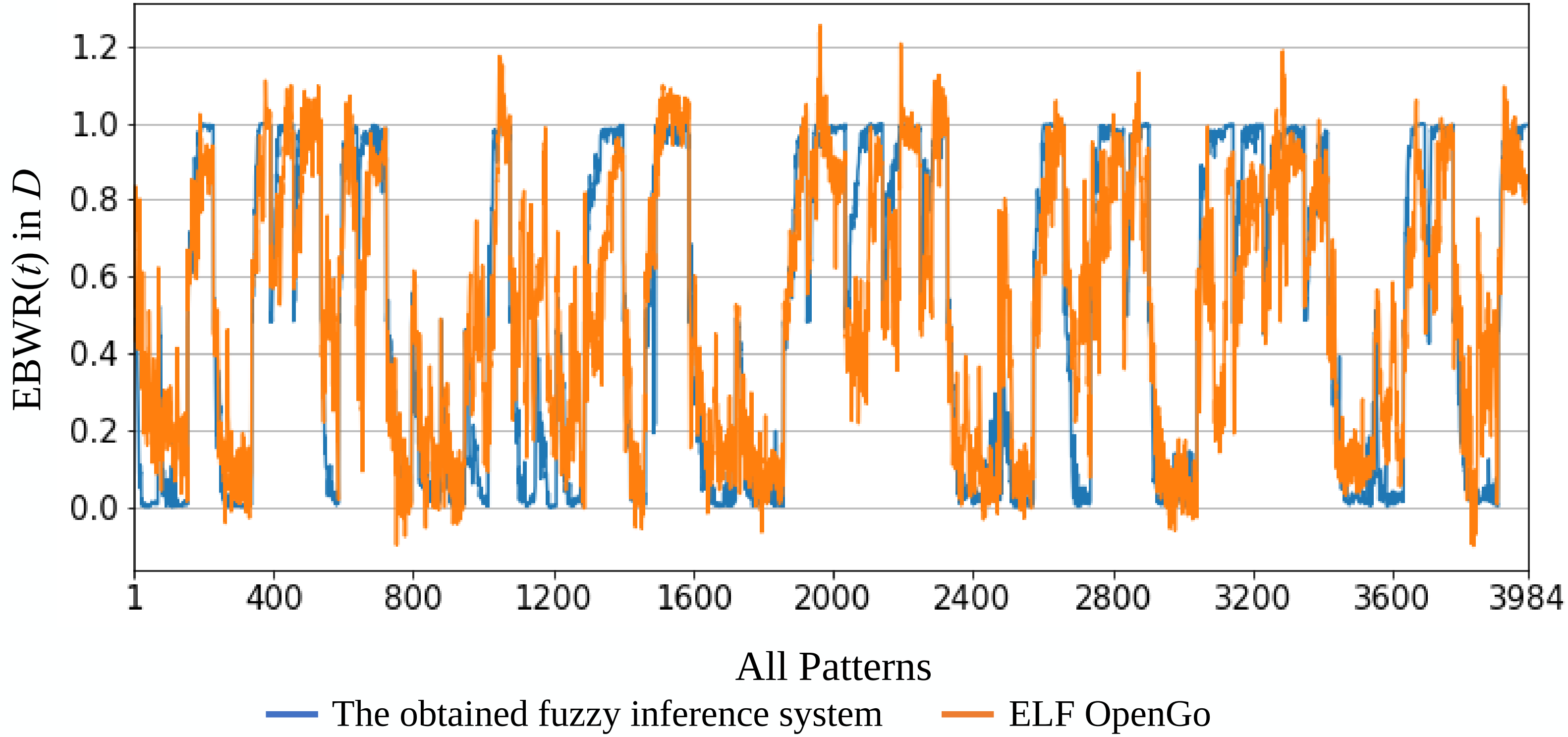
- This figure shows that inferred DBWR(t+1) curves for the test dataset by our proposed method and ELF OpenGo AI.



Computational Experiments

Inferred EBWR(t) Curves for the Training Dataset

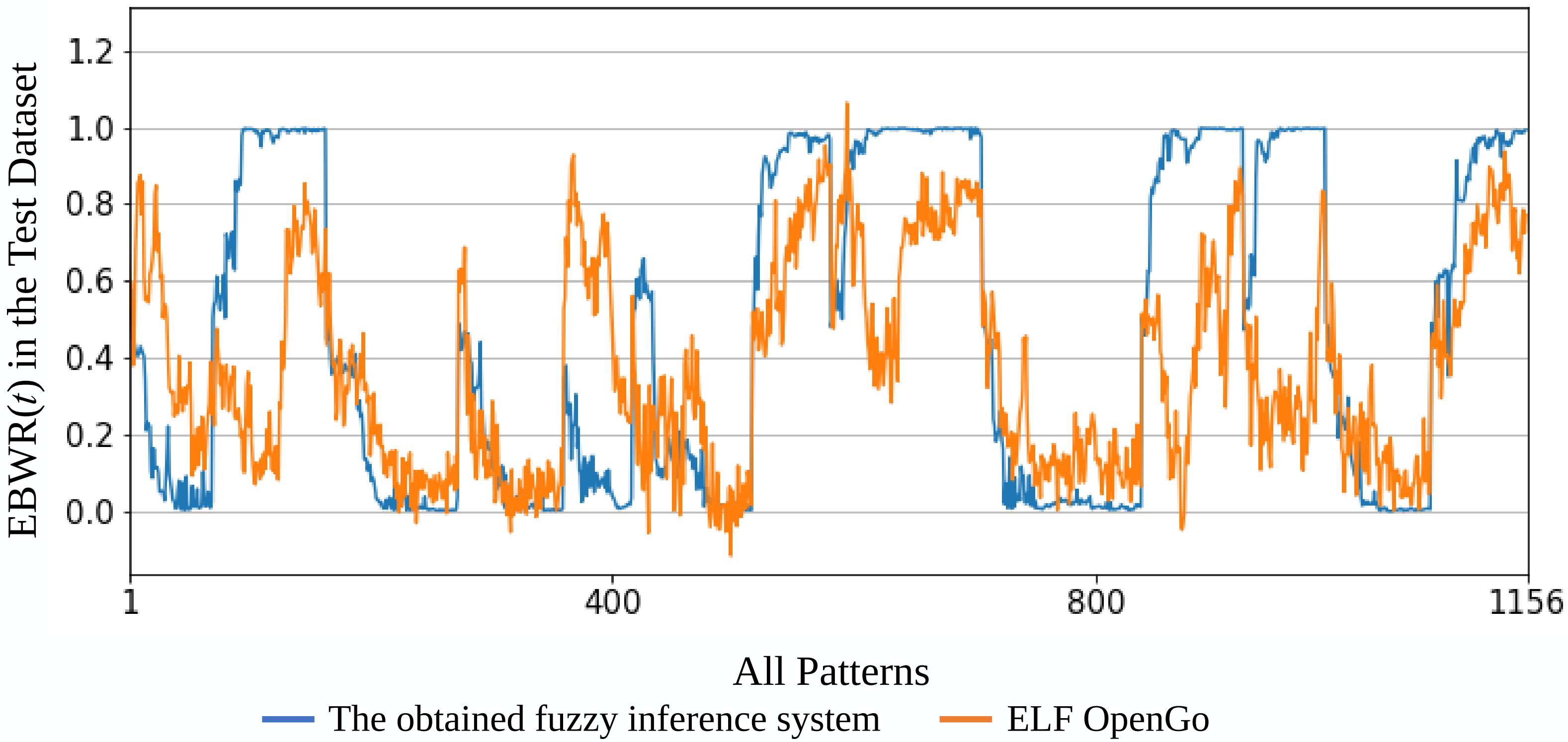
- This figure shows that inferred EBWR(t) curves for the training dataset by our proposed method and ELF OpenGo AI.
- Note that the training dataset includes the learning set and the evaluation set.



Computational Experiments

Inferred EBWR(t) Curves for the Test Dataset

- This figure shows that inferred EBWR(t) curves for the test dataset by our proposed method and ELF OpenGo AI.



Computational Experiments

Selected Features and MSEs for DBWR(t+1)

- We show the selected features for desired output variables and the MSE values by fuzzy inference systems using the selected features.

Feature Array	MSE Training	MSE Test	Best: Red Validation
000000000100	1.15e-03	7.99e-04	1.21e-03
000100000100	9.49e-04	6.17e-04	1.02e-03
000100001100	**8.77e-04**	**5.76e-04**	**9.71e-04**
010100001100	9.05e-04	6.64e-04	1.05e-03
010110001100	8.88e-04	6.39e-04	9.97e-04
010110011100	9.15e-04	6.50e-04	1.04e-03
010110011110	8.86e-04	6.25e-04	9.95e-04
010110011111	9.40e-04	6.80e-04	1.07e-03
011110011111	9.04e-04	6.25e-04	1.00e-03
011110111111	9.47e-04	7.03e-04	1.07e-03
011111111111	9.33e-04	6.55e-04	1.04e-03
111111111111	9.81e-04	7.12e-04	1.13e-03

Computational Experiments

Selected Features and MSEs for EBWR(t)

- We show the selected features for desired output variables and the MSE values by fuzzy inference systems using the selected features.

Feature Array	MSE Training	MSE Test	MSE Validation (Best: Red)
100000000000	1.77e-01	1.82e-01	1.73e-01
100000000100	6.52e-02	**9.16e-02**	6.50e-02
100000000101	5.55e-02	1.10e-01	5.77e-02
100000000111	4.79e-02	1.03e-01	5.08e-02
101000000111	4.67e-02	1.08e-01	4.93e-02
111000000111	4.86e-02	1.29e-01	5.17e-02
111000100111	**4.62e-02**	1.18e-01	5.01e-02
111100100111	4.66e-02	1.19e-01	5.08e-02
111100110111	4.76e-02	1.15e-01	5.08e-02
111101110111	4.74e-02	1.27e-01	5.00e-02
111111110111	4.71e-02	1.18e-01	**4.89e-02**
111111111111	4.84e-02	1.07e-01	4.90e-02

Conclusions

Conclusions

- We proposed an evolutionary FML-based fuzzy inference system with a feature selection method.
- The proposed method selected effective features and showed high generalization ability for estimating the future DBWR.

Future Research Topics

- Consideration of a better way to select a set of effective features for minimizing MSE value for the test dataset.